

什么是http协议?

http协议 (超文本传输协议 hyper text transfer protoclo)

所谓协议, 就是指双方遵循的规范 (浏览器和服务端之间沟通的一种规范)

HTTP是一个 (基于TCP/IP) 应用层的面向对象的协议, 由请求和响应构成, 是一个标准的客户端服务器模型。适用于分布式超媒体信息系统

浏览网页是HTTP的主要应用, 但是这并不代表HTTP就只能应用于网页的浏览。HTTP是一种协议, 只要通信的双方都遵守这个协议, HTTP就能有用武之地。比如咱们常用的QQ,迅雷这些软件, 都会使用HTTP协议 (还包括其他的协议)。

http协议的特点

- 1.HTTP默认的端口号为80, HTTPS的端口号为443。
- 2.支持B/S及C/S模式(客户服务器模式/浏览器/服务器模式)
3. 简单快速: 客户向服务器请求服务时, 只需传送请求方法和路径。由于HTTP协议简单, 使得HTTP服务器的程序规模小, 因而通信速度很快。
- 4.常用的请求方式有GET,POST,HEAD
- 5.灵活: http运行传输任意类型数据对象
- 6.无连接: 无连接的含义是限制每次请求只处理一个请求, 服务器处理完客户的请求, 并收到客户的应答后, 即断开连接, 采用这种方式以节省传输时间。
- 7.无状态: HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息,则它必须重传,这样可能导致每次连接传送的数据量增大。另一方面, 在服务器不需要先前的信息时它的应答比较快

http的工作原理:

http协议定义web客户端如何从web服务器请求web页面, 以及服务器如何把web页面传输给客户端, http采用了请求/响应模式, 客户端向服务器发送一个请求报文, 请求报文中包含了请求方法, url (请求地址), 协议版本, 请求头部, 请求数据, 服务器以一个状态行为作为响应, 响应的内容包含协议的版本, 请求成功/失败的代码, 服务器的信息, 响应头部和响应的数据。

请求过程:



请求过程的特点:

- 1.只能是客户端主动发起请求
- 2.所有的资源（文件）都需要发起一个请求
- 3.所有的资源都只是从服务区加载到本地，然后在客户端执行
- 4.在执行的过程中如果需要加载资源就继续发起网络请求

一次请求的四个步骤:

- 1.建立连接open
- 2.客户端像服务器发起请求 send.request
- 3.服务器响应数据到客户端
- 4.断开连接

请求方式:

1.GET

GET方法的作用在于获取资源，轻便高效，有传值的大小限制，不是http协议造成的是浏览器的问题但是安全性比较低，而且不能用于文件提交，文件提交只能用post

2.POST

POST方法的主要作用是执行操作,主要用于提交数据，传输的数据在请求体里面，提交数据没有大小限制，这个和服务器端的配置有关，安全性比较高。

3.HEAD

HEAD这个方法的功能与GET方法相似，不同之处在于服务器不会在其响应中返回消息主体。服务器返回的消息头应与对应GET请求返回的消息头相同。因此，这种方法可用于检查某一资源在向其提交GET请求前是否存在

post和get的区别:

- 1.get请求只能进行url编码，而post支持多种编码
- 2.get请求在url中传参是有长度限制的，post没有

- 3.get方法是通过url传递参数的，所以产生可能会有一些特殊的字符例如 '+'，浏览器对其进行编码可能会造成后台报错，而post不会
- 4.同源的时候，get请求里面会带有origin字段，而post不会
- 5.GET 参数通过 URL 传递，POST 放在 Request body 中。
- 6.GET 比 POST 更不安全，因为参数直接暴露在 URL 上，所以不能用来传递敏感信息

常见的状态码：

- 100 Continue当客户端提交一个包含主体的请求时.将发送这个响应。该响应表示已收到请求消息头.客户端应继续发送主体。请求完成后，再由服务器返回另一个响应。
- 200 OK本状态码表示已成功提交请求，且响应主体中包含请求结果。
- 201 Created. PUT请求的响应返回这个状态码，表示请求已成功提交。
- 301 Moved Permanently本状态码将浏览器永久重定向到另外一个在Location消息头中指定的 URL。以后客户端应使用新URL替换原始URL。
- 302 Found本状态码将浏览器暂时重定向到另外一个在Location消息头中指定的URL.客户端 应在随后的请求中恢复使用原始URL.
- 304 Not Modified本状态码指示浏览器使用缓存中保存的所请求资源的副本。服务器使用 If-Modified-Since与If-None-Match消息头确定客户端是否拥有最新版本的资源。
- 400 Bad Request本状态码表示客户端提交了一个无效的HTTP请求。当以某种无效的方式修改请求时（例如在URL中插入一个空格符），可能会遇到这个状态码
- 401 Unauthorized.服务器在许可请求前要求HTTP进行身份验证。WWW-Authenticate消息头详细说明所支持的身份验证类型。
- 403 Forbidden本状态码指出，不管是否通过身份验证，禁止任何人访问被请求的资源。
- 404 Not Found本状态码表示所请求的资源并不存在。
- 405 Method Not Allowed本状态码表示指定的URL不支持请求中使用的方法。例如，如果试图在不支持PUT方法的地方使用该方法，就会收到本状态码。
- 413 Request Entity Too Large 则本状态码表示请求主体过长，服务器无法处理。
- 414 Request URL Too Long与前一个响应类似，本状态码表示请求中的URL过长，服务器无法处理。
- 500 Internal Server Error本状态码表示服务器在执行请求时遇到错误。当提交无法预料的输入、在应用程序处理过程中造成无法处理的错误时，通常会收到本状态码。应该仔细检查服务器响应的所有内容，了解与错误性质有关的详情。
- 503 Service Unavailable通常，本状态码表示尽管Web服务器运转正常.并且能够响应请求,但服务器访问的应用程序还是无法作出响应。应该进行核实，是否因为执行了某种行为而造成这个结果。

tcp：三次握手四次挥手

第一次握手：客户端给服务器端发送一个syn报文，并指明客户端的初始化序列号ISN

第二次握手:服务器收到客户端的syn报文后, 会以自己的syn报文作为应答并且指定了自己的初始化序列号,

第三次握手: 客户端收到syn报文后会发送一个ACK报文, 把服务器的ISN+1作为ASK的值, 表示已经收到了服务器端的syn报文, 此时客户端处于ESTABLISHED状态 (表示两台机器正在传输数据)

tcp: 三次握手四次挥手

第一次挥手: 客户端会发送一个FIN报文, 报文中会指定一个序列号, 此时客户端处于FIN-WAIFI状态 (则表明是服务端主动发起的断开)

第二次挥手:服务器收到FIN之后, 会发送ACK报文, 且客户端的序列号+1作为ACK报文的序列号值, 表明已经收到客户端的报文了, 此时服务器端处于CLOSE-WAIT状态 (被动关闭)

第三次挥手: 如果服务器也想断开连接了, 和客户端的第一次挥手一样, 会发生FIN报文, 且指定一个序列号, 此时服务器处于LAST-ACK的状态 (当被动关闭的一方在发送FIN报文后, 等待对方的ACK报文的时候, 就处于LAST_ACK的状态)

第四次挥手: 客户端收到FIN报文之后, 一样会发送一个ACK报文作为应答, 且把服务器端的序列号值+1作为自己的ACK报文的序列号值, 此时客户端处于TNME-WAIT状态 (意思是结束了这次连接。)

扩展: https协议

HTTPS协议作为TCP/IP模型中应用层的协议也不例外。HTTP协议通常承载于TCP协议之上, 有时也承载于TLS或SSL 协议层之上, 这个时候, 就成了我们常说的HTTPS。